

Parallel Stroked Multi Line: a model-based method for compressing large fingerprint databases

Hamid Mansouri^{a,*}, Hamid-Reza Pourreza^{a,**}

^a*Machine Vision Lab., Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad 91779, Iran*

Abstract

With increasing usage of fingerprints as an important biometric data, the need to compress the large fingerprint databases has become essential. The most recommended compression algorithm, even by standards, is JPEG2000. But at high compression rates, this algorithm is ineffective. In this paper, a model is proposed which is based on parallel lines with same orientations, arbitrary widths and same gray level values located on rectangle with constant gray level value as background. We refer to this algorithm as Parallel Stroked Multi Line (PSML). By using Adaptive Geometrical Wavelet and employing PSML, a compression algorithm is developed. This compression algorithm can preserve fingerprint structure and minutiae. The exact algorithm of computing the PSML model take exponential time. However, we have proposed an alternative approximation algorithm, which reduces the time complexity to $O(n^3)$. The proposed PSML algorithm has significant advantage over Wedgelets Transform in PSNR value and visual quality in compressed images. The proposed method, despite the lower PSNR values than JPEG2000 algorithm in common range of compression rates, in all compression rates have nearly equal or greater advantage over JPEG2000 when used by Automatic Fingerprint Identification Systems (AFIS). At high compression rates, according to PSNR values, mean EER rate and visual quality, the encoded images with JPEG2000 can not be identified from each other after compression. But, images encoded by the PSML algorithm retained the sufficient information to maintain fingerprint identification performances similar to the ones obtained by raw images without compression. One the *U.are.U 400* database, the mean EER rate for uncompressed images is 4.54%, while at 267:1 compression ratio, this value becomes 49.41% and 6.22% for JPEG2000 and PSML, respectively. This result shows a significant improvement over the standard JPEG2000 algorithm.

Keywords: Parallel stroked multi line, Adaptive geometrical wavelet,

*Corresponding author, Tel: +98 915 509 8244

**Principal corresponding author, Tel: +98 51 3880 5025

Email addresses: mansouri@stu.um.ac.ir (Hamid Mansouri), hpourreza@um.ac.ir (Hamid-Reza Pourreza)

1. Introduction

Biometric-based human identification and authentication systems rely on several biometric modalities including Fingerprint [1], Palm [2], Face [3], Iris [4], Gait [5] and human activity [6]. Among these modalities, fingerprint is the most widely used one. Traditionally fingerprint data used to be stored on paper. But, with advancements in machine-based fingerprint analysis, the data is now stored in digital format. Automated Fingerprint Identification Systems (AFIS) utilize the digitized data to perform the matching and verification. These systems are now highly reliable and even European Union (EU) recently decided to use fingerprint data in digital passports [7]. With rapid growth of usage of fingerprint-based systems, the size of fingerprint enrollment databases grows significantly. Therefore, efficient storage mechanisms are required to handle this massive amount of data. Also, in distributed biometric systems, sensors are physically located in a location different from the identification system itself. In most cases, fingerprint data are transmitted from sensor to processing system with low bandwidth and high latency wireless links. So, sensor data compression is necessary.

Two solutions can be offered to alleviate the data size problem. First, storing only the required features for identification (e.g., minutiae) instead of the raw data, which takes much less space as opposed to the raw data. Second solution is compressing the raw data. Since in some cases the algorithm may evolve over time, we may need to extract new features, which are not captured by the original feature set. Hence, the raw image is needed to perform enrollment phase again. Therefore, storing the compressed raw image is more desirable.

Compressing fingerprint data can be applied in two forms, Lossy or Lossless. Lossless compression can achieve up to 4:1 compression ratio, which may not yield sufficient storage size reduction [7]. The most widely used lossless compression formats include PNG, GIF, JPEG Lossless, JPEG-LS and JPEG2000 Lossless. Lossy compression can achieve desired compression rate for storing fingerprint image data. But, the compression may introduce distortion on the fingerprint images. This distortion can affect AFIS identification rate and decrease their performances. Compression distortion is usually measured by peak signal-to-noise ratio (PSNR) metric (i.e., the higher the PSNR, the higher quality of the compression). However, PSNR may not directly translate into the best quality metric for AFIS system as different distortions may affect the extracted minutiae information differently. Therefore, traditional quality metrics are not used on fingerprint data; instead of them, measures that are obtained from end-to-end matching and identification results of AFIS are often used. These metrics include False Acceptance Rate (FAR), False Rejection Rate (FRR) and Equal Error Rate (EER) are used [8].

Although standards recommend specific compression algorithms to apply on fingerprint data, there exists other specialized algorithms targeting these type of images. Some of these algorithms used the special structure that exists in fingerprint images for compression. In [9] ridge structure is considered for compressing images. By focusing on similarity between structures bounded in small patches, [10] used Compressed Sensing (CS) for compression. This method requires training phase and considering the amount of fingerprint images in large datasets, performance of the algorithm might be reduced. In this paper, an algorithm proposed by focusing on special structure called Parallel Stroked Multi Line (PSML). PSML refers to a structure that contains multiple parallel lines and could be drawn as a parametric geometric model. So, in comparison to CS methods, this kind of representation does not require pre-trained dictionary for encoding and decoding. The proposed method is comparable with JPEG2000 in common range of compression ratio, But it has a better performance in low compression rates (higher quality) and very high compression rates (very low quality).

The rest of the paper is organized as follows. The related works are discussed in section 2. Before presenting the proposed algorithm, model and related parameters are described in section 4, we first review the base algorithm (i.e., adaptive geometrical wavelet) in section 3. Section 5 presents with the details of the parameters quantization and model computation, which also presents our fast model computation algorithm. Section 6 provides the result of experiments and highlights the advantages of the proposed algorithm with evaluations on standard databases. Section 7 concludes the paper and suggests future directions of this work.

2. Previous works

By developing the usage of biometric data in last two decades, several algorithms and standards were set up for compressing data in biometric systems. The most relevant one ISO/IEC 19794 standard was developed for Biometric Data Interchange Format, where Part 4 focused on fingerprint data. ISO/IEC 19794-4 allows fingerprint images data to be stored in lossy manner in JPEG[11], wavelet transform/scalar quantization (WSQ)[12] and JPEG2000[13, 14] formats, where the later is recommended. Also [15] proposed a corresponding specific JPEG2000 Part I profile for 1000 ppi fingerprint images.

While using the data formats specified by the ISO/IEC 19794-4 standard established in most applications, the lack of recognition accuracy in comparison with other compression algorithms has led to development of new algorithms or using the general purpose ones for compressing fingerprint images. In general, the algorithms are divided in two categories, lossless or lossy, according to type of compression. But, this paper summarizes the previous works into four categories, which are discussed in the following.

The first category includes researches that review, compare and analyze existing methods proposed for compressing fingerprint images, any type of images, or any type of data. In [16] a complete comparison between JPEG and WSQ

is done that shows WSQ have a better compression ratio against JPEG for fingerprint images. Based on the strong texture structure in fingerprint images, [17] preformed a comparison between Wavelet Transform and Fractal Coding methods for texture-based images. Compression methods could be used together as a hybrid one, so [18] compared standard types of Wavelet, Fractal and JPEG. In addition, a comparison among standard types and hybrid types was performed. Later, due to the need of higher compression ratio for fingerprint images, [19] performed a comparison among JPEG, JPEG2000, EZW and WSQ at higher compression ratio (40:1) and concluded that JPEG2000 has better performance than WSQ in higher compression ratio. This result was confirmed by [20] with PSNR and Spectral Image Validation and Verification (SIVV) [21] metrics. In addition to above methods, [7] used set partitioning in hierarchical trees (SPIHT) and predictive residual vector quantization (PRVQ) compression methods in comparison. A fully comparison between lossless compression methods is done in [22] that includes known methods like Lossless JPEG, JPEG-LS, Lossless JPEG2000, SPIHT, PNG, GIF and global compression methods like Huffman Coding, GZ, BZ2, 7z, RAR, UHA and ZIP. The metrics used for this comparison was FAR and FRR.

As stated above, methods based on Discrete Cosine Transform (DCT) and Wavelet Transform have most usage on fingerprint image compression. Also, coefficients related to higher frequency have more significance in fingerprint images rather than public images. Therefore, second category includes researches that focus on energy and coefficient distribution on these transforms. [23, 24] used Simulated Annealing (SA) to optimize wavelets for fingerprint image compression by searching the space of wavelet filter coefficients to minimize an error metric. The distortion metric used by [24] is a combination of objective RMS error on compressed images and subjective evaluations by fingerprint experts. [25, 26] proposed a similar approach by using Genetic Algorithm (GA) instead of SA. In [26] fingerprint image divided into small patches (e.g. 32x32), then used GA to optimize the wavelet coefficients over them. Their evaluations show that the learning time was decreased while PSNR increased. Kasaei and et al. [27, 28, 29] used Piecewise-Uniform Pyramid Lattice Vector Quantization to propose a method based on generalized Gaussian distribution for the distribution of the wavelet coefficient. Different filter banks that could be used in wavelet transform were reviewed in [30] and the conclusion is that using Coif 5 filter bank is preferred to Bior 7.9 filter bank, that is used in standard wavelet transform, producing higher compression ratio in fingerprint images. As an alternative strategy, special energy distribution based on fingerprint pattern can be used for determining coefficients. [31] used this strategy for DCT-based coders.

The third category includes researches focused on structures founded in fingerprint images and used these structures for compression. The most prominent structure that can be found, is *ridge*. Another structure-based compression is *valley*. [32] used Ridgelet Transform and [33] used hybrid model from ridge and valley for compressing fingerprint images. The existence of these strong structures has led researchers to use Sparse Representation (SR). For instance

[34, 35] uses Self Organizing Map (SOM) – is kind of SR – to represent patches of fingerprint images in a compressed format. [34] used SOM and Learning Vector Quantization (LVQ) to learn the network, while [35] used Wave Atom Decomposition and SOM. Also, k-SVD [10] is used for sparse representation of image patches. This method could achieve better performance against JPEG2000 and WSQ with PSNR metric.

The last category includes researches that could not be categorized in other categories. These researches provide new methods for compressing fingerprint images. In [36] a method is proposed based on Directional Filter Banks and Trellis Coded Quantization (TCQ). This method takes Wavelet-based Contourlet Transform from fingerprint image, then using TCQ, encodes the resulting coefficients and produces a compressed result. The results showed an improvement in performance over SPIHT. As mentioned before, lossless compression can also be used for compression. [37] proposed a lossless compression method by changing the prediction function. The fingerprint image was divided into four parts, then the prediction function of top-right part changed to use information provided by itself and mirrored top-left part.

3. Adaptive Geometrical Wavelet

From human vision system, it is known that the human eye is designed to catch changes of location, scale and orientation [38]. Classical wavelets were efficient in catching location and scale, but not efficient in catching orientation. Despite development of classical wavelet into 2D space [39, 40, 41], for efficient catching of orientation, Geometrical Wavelet was proposed. Geometrical wavelet can be divided into two groups, nonadaptive and adaptive. The first group is based on nonadaptive methods of computing, which use frames like Brushlets [42], Ridgelets [43], Curvelets [44], Contourlets [45] and Shearlets [46]. The second group approximates image in an adaptive way. Most of them are based on dictionary, like Wedgelets [47], Beamlets [48], Second-order Wedgelets [49], Platelets [50, 51] and Surflets [52]. Recently, some approaches based on basis were proposed, like Bandelets [53], Grouplets [54] and Tetrolets [55].

In remaining of this section, basis definitions common to all adaptive geometrical wavelets are expressed. Then, the most known member of this family, Wedgelets, is briefly introduced.

3.1. The Class of Horizon Functions

Let us define image domain $S = [0, 1] \times [0, 1]$. Function $h(x)$ defined in S is called *horizon*, if it is continuous, smooth, defined on interval $[0, 1]$ and it fulfills the Hölder regularity condition. In practice, it is sufficient h be a member of C^2 class. Assume characteristic function

$$H(x_1, x_2) = \mathbf{1}_{\{x_2 \leq h(x_1)\}} \quad 0 \leq x_1, x_2 \leq 1.$$

Function H is called a *horizon function*, if h is horizon. H formed a black and white image that under the horizon is black and upper it is white. An example of horizon and horizon function is shown in Fig. 1.

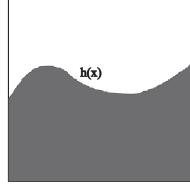


Figure 1: An example of horizon and horizon function [47]

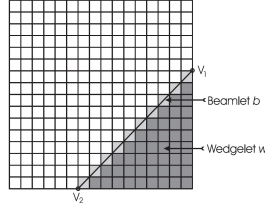


Figure 2: An example of wedgelet defined by beamlet b [47]

3.2. Dictionary of Wedgelets

Denote dyadic square $S(j-1, j_2, i)$ defined as 2D range

$$S(j_1, j_2, i) = [j_1/2^i, (j_1 + 1)/2^i] \times [j_2/2^i, (j_2 + 1)/2^i]$$

that $0 \leq j_1, j_2 < 2^i$; $i \geq 0$; $j_1, j_2, i \in \mathbb{N}$. Let's define image with $N \times N$ pixels. If assume $N = 2^I$ then $S(0, 0, 0)$ denote the whole image domain and $S(j_1, j_2, I)$ denote all pixels of image. In each border of square $S(j_1, j_2, i)$ there exist vertices with distance equal to $\frac{1}{N}$. Every pair of such vertices can be connected to form a straight line – edge (also called *beamlet* after work of Donoho and Hou [47]). Therefore, one can define edges in different location, scale and orientation. The set of these edges forms a binary dictionary. Each edge b takes square S into two parts. Let us define one of the parts that is bounded between edge and up right corner with following indicator function

$$w(x_1, x_2) = \mathbf{1}_{\{x_2 \leq b(x_1)\}}$$

Such function is called *wedgelet*, and is defined by beamlet b . The graphical representation of wedgelet defined by beamlet b is shown in Fig. 2.

The set of wedgelets in any S can be defined as

$$W(S) = \{\mathbf{1}(S)\} \cup \{\text{all possible } w \text{ defined on } S\}.$$

From now, assume the subscripts j_1, j_2 such that $0 \leq j_1, j_2 < 2^i$ is replaced by subscript j such that $0 \leq j < 4^i$. Also lets denote square $S(j_1, j_2, i)$ as $S_{i,j}$. Also, lets denote parametrization of orientation with m , which was denoted coordinate v_1, v_2 previously.

Definition 1. The *Wedgelets Dictionary* are defined as the following set

$$W = \{w_{i,j,m} : i = 0, \dots, \log_2 N ; j = 0, \dots, 4^i - 1 ; m = 0, \dots, M_W(S_{i,j}) - 1\} \quad (1)$$

where $M_W(S_{i,j})$ denotes the number of wedgelets in $S_{i,j}$.

3.3. Wedgelet Transform

Consider image as $F : S \rightarrow \mathbb{R}$, so the wedgelets transform is defined as following

Definition 2. *Wedgelets Transform* defined as following formula

$$\alpha_{i,j,m} = \frac{1}{T} \int \int_S F(x_1, x_2) w_{i,j,m}(x_1, x_2) dx_1 dx_2$$

where

$$T = \int \int_S w_{i,j,m}(x_1, x_2) dx_1 dx_2$$

is the normalization factor and $S = [0, 1] \times [0, 1]$, $\alpha_{i,j,m} \in \mathbb{R}$, $w_{i,j,m} \in W$, $0 \leq i \leq \log_2 N$, $0 \leq j < 4^i$, $0 \leq m < M_W(S_{i,j})$ and $i, j, m \in \mathbb{R}$.

In grayscale images, coefficients quantized to $\alpha_{i,j,m} \in \{0, \dots, 255\}$, while in binary images coefficients are quantized to $\alpha_{i,j,m} \in \{0, 1\}$. Finally, image representation with wedgelets is defined as following:

$$F(x_1, x_2) = \sum_{i,j,m} \alpha_{i,j,m} w_{i,j,m}(x_1, x_2)$$

But, because W is dictionary, not basis, not all of coefficients in above formula are used [49] (It means some of coefficient are set to zero). At last, we are looking for the best approximation image using minimum number of atoms from a given dictionary.

3.4. Wedgelet Analysis of Image

Almost all multiresolution methods, including wedgelet approximation, use quadtree as main data structure. Although many ways exist to store wedgelets with the help of quadtree, but the most common way is the one that assumes in each node of quadtree, the coefficients determining the appropriate wedgelet are stored.

The basic algorithm of decomposition image, contains two steps. In first step, full image decomposition using wedgelets transform is obtained. It means. for each square S , the best approximation with minimum square error using wedgelet is obtained. In second step, a bottom-up optimization pruning algorithm is applied on decomposition tree to obtain best quality approximation

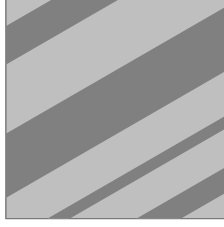


Figure 3: An example of the PSML model

image with minimum number of atoms. Indeed, the following weighted formula is minimized [47]:

$$R_\lambda = \min_P \{ \|F - F_W\|_2^2 + \lambda^2 K \},$$

where P is homogeneous partition of an image, F denotes original image, F_W is wedgelet approximation, K is the number of bits required to encode approximation and λ is rate distortion parameter.

4. Proposed method

Fingerprint images can be viewed simply as black and white images. In this form of images, fingerprint images seem to be made of by black lines mashing around themselves in white background. If a small patch of image is considered, a couple of parallel black lines are seen. So, this aspect of view, represents a geometric model that is built up with parallel lines with arbitrary widths, named *parallel stroked multi lines* model. In this paper, an algorithm based on Adaptive Geometrical Wavelet is used to employ PSML model for compressing fingerprint images. in the rest of this section, the proposed model and its parameters are described in details.

4.1. Parallel Stroked Multi Line model

Parallel Stroked Multi Line model is built up by some parallel lines. Each line may have an arbitrary width. The gray level value of all pixels lie on each line is the same. Likewise, the gray level value of all background pixels, between lines, are the same (see Fig. 3).

Definition 3. Assume rectangle S with arbitrary size and parameters θ , k , c_1 and c_2 . Suppose part of k lines l_1 to l_k with width w_{l_1} to w_{l_k} respectively and slope θ that lies in S . By considering c_1 as gray level value of lines and c_2 as the value of S gray level, rectangle S defines a gray level image. This image forms Parallel Stroked Multi Line model.

4.2. Model parameters

For simplicity of model representation, each line l_i with width w_{l_i} is assumed as w_{l_i} lines with unit width. The background part between each two lines (l_i and l_{i+1}) also assumed as some lines with unit width. Thus, the proposed model can be represented by a set of parallel lines with unit width, slope θ and gray level value c_1 or c_2 . Considering the rectangle S is made by p lines with unit width and slope θ , then the gray level value of all lines in S can be represented by binary stream $G = g_1 g_2 \cdots g_p$ (zero means c_1 and one means c_2). Therefore, PSML can be represented with parameters θ , c_1 , c_2 and G .

5. Model Computation

To compute the model, first parameter quantization is described. After that, time complexity of computation is analyzed, then, an approximation method is proposed to speed up the proposed algorithm. Finally, a simple compression algorithm proposed to compress the resulting coefficients by the PSML algorithm.

5.1. Model parameters quantization

In this section, parameter quantization is described, separately. Parameter θ varies in range $[0, \pi)$ to cover all directions. For this purpose two points C and O were used to represent orientation. The angle between the line passing from points C and O and the horizontal axis of coordinate system, forms θ . Suppose that square S is $m \times n$. Point C is fixed and located at pixel $C = (\lfloor \frac{m}{2} \rfloor, \lceil \frac{n}{2} \rceil)$. Location of point O varies and can be any pixel in set $D_O = \{(0, y) | y = 0, \dots, n-1\} \cup \{(x, n-1) | x = 0, \dots, m-1\}$. Thus, the number of difference state of parameter θ is equal to $m+n-1$.

The value of gray level parameters c_1 and c_2 are usually represented by 8 bits. But, if needed, it can be reduced for better compression. Finally, binary stream G is coded maximumly with $\sqrt{m^2 + n^2}$ bits.

5.2. Time complexity of model computation

Theorem 1. *The time complexity of approximating $n \times n$ patch with Parallel Stroked Multi Line and using MSE metric is $O(n^3 2^n)$.*

PROOF. Each patch with size $m \times n$ has $(m+n-1)2^{\sqrt{m^2+n^2}}$ different structural states. For $n \times n$ patch approximately $O(n 2^n)$ different structural states exist. In each of these states, the best value for c_1 (or c_2) is computed by mean of gray level of pixels that have value zero (or one) in binary stream G . So, to calculate the best approximation of $n \times n$ patch, all of these states should be computed and the one with lowest MSE is chosen. The calculating the mean values in each patch needs $O(n^2)$ operations. Therefore, best approximation of a patch has $O(n^3 2^n)$ time complexity.

5.3. Fast patch approximation algorithm

Since the time complexity of approximation of each patch has exponential order, an approximation method is proposed to compute it faster. The proposed method is based on *Hill Climbing* algorithm and finds the local maximum solution. In this method, parameter θ is known and we try to find parameter G . After this parameter is computed, parameters c_1 and c_2 could easily be computed.

Algorithm 1 (Fast Approximation). *Definition:* Denote binary stream $G = g_1g_2 \cdots g_p$. Every binary stream G_k ($k = 1, \dots, p$) is defined as

$$G_k = g'_1g'_2 \cdots g'_p \quad \begin{cases} g'_j = g_j & j \neq k \\ g'_j = \text{not}(g_j) & j = k \end{cases}$$

and is called neighbors of binary stream G . Let's denote all neighbors of state G with $N(G)$.

Initiation: Assume initial state as G^0 and current state as G^c . To initialize the algorithm set $G^c = G^0$ and $i = 1$.

Selection: In each step i of algorithm, the neighbor $G^i \in N(G^c)$ with lower approximation error than all states in $N(G^c)$, is selected. If $\text{err}_{G^i} < \text{err}_{G^c}$, set $G^c = G^i$ and do the selection step again with $i = i + 1$. If $\text{err}_{G^i} \geq \text{err}_{G^c}$ then $G^r = G^i$ and go to the next step.

Output: The best approximation with lowest error of patch with known parameter θ is G^r .

The flowchart of this algorithm is depicted in Fig. 4.

From the experiments, the best initial state that produces result near the optimal, is generated by the following algorithm.

Algorithm 2 (Initial State). Denote the mean of all gray level value of all pixels in the image as c_m . For each line l_i ($i = 1, \dots, p$), If the mean of gray level value of all pixels of the line is lower than c_m , then let $g_i = 1$, else $g_i = 0$. The produced binary stream forms the initial state.

Corollary 1. From the experimental results, the selection step of algorithm 1 is repeated in order of $O(n)$.

Lemma 1. The best approximation of each $n \times n$ patch with PSML and fast approximation algorithm is computed in $O(n^5)$.

PROOF. The approximation algorithm for each $n \times n$ patch is computed with following algorithm:

- | | | |
|---|--|----------------------|
| 1 | Do for each orientation θ | $O(n)$ |
| 2 | Compute initial state | $O(n^2)$ |
| 3 | Repeat the selection phase | $O(n)$ (corollary 1) |
| 4 | Do for each neighbor in $N(G^i)$ | $O(n)$ |
| 5 | Calculate c_1 and c_2 parameters for G_k | $O(n^2)$ |

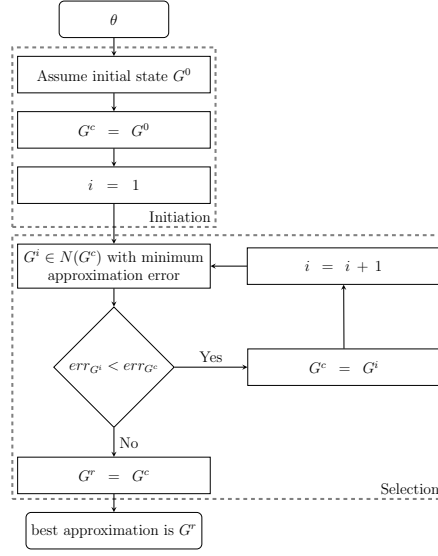


Figure 4: The flowchart of Algorithm 1 (Fast Approximation) that compute patch parameters with fast approximation algorithm.

Accordingly, the best approximation of each $n \times n$ patch is computed in $O(n^5)$.

Let's suppose each line with unit width in rectangle S is denoted by u_i ($i = 1, \dots, p$). Sum of pixel gray level values and number of pixels belonging to each line are denoted by N_i and S_i , respectively. Then, the c_1 and c_2 parameters are calculated as

$$c_1 = \frac{\sum_{i=1, \dots, p} S_i}{\sum_{i=1, \dots, p} N_i}, \quad c_2 = \frac{\sum_{i=1, \dots, p} S_i}{\sum_{i=1, \dots, p} N_i}.$$

This form of computation reduces the time complexity of calculating c_1 and c_2 parameters to $O(n)$. Surprisingly, this time complexity can reduce much more according to the following lemma.

Lemma 2. *The time complexity of calculating c_1 and c_2 parameters for $G_k \in N(G)$ is $O(1)$.*

PROOF. Let's define $SC_{c,G}$ and $NC_{c,G}$ as

$$SC_{c,G} = \sum_{i=1, \dots, p} S_i, \quad NC_{c,G} = \sum_{i=1, \dots, p} N_i.$$

The parameters c_1^k and c_2^k of G_k can be computed as

$$\begin{cases} \text{if } g_i = 0 & c_1^k = \frac{SC_{0,G} - S_k}{NC_{0,G} - N_k}, \quad c_2^k = \frac{SC_{1,G} + S_k}{NC_{1,G} + N_k} \\ \text{if } g_i = 1 & c_1^k = \frac{SC_{0,G} + S_k}{NC_{0,G} + N_k}, \quad c_2^k = \frac{SC_{1,G} - S_k}{NC_{1,G} - N_k} \end{cases},$$

that takes $O(1)$ time complexity. Also, the parameters SC_{c,G^k} and NC_{c,G^k} can be computed in $O(1)$.

By using the lemma 2, the approximation algorithm for each $n \times n$ patch is computed with following algorithm:

- | | | |
|---|--|----------------------|
| 1 | Do for each orientation θ | $O(n)$ |
| 2 | Compute initial state $(S_i, N_i, SC_{c,G^0}, NC_{c,G^0})$ | $O(n^2)$ |
| 3 | Repeat the selection phase | $O(n)$ (corollary 1) |
| 4 | Do for each neighbor in $N(G^i)$ | $O(n)$ |
| 5 | Calculate c_1 and c_2 parameters for G_k | $O(1)$ (lemma 2) |

Theorem 2. *Time complexity of the fast approximation algorithm used to obtain the best approximation of the whole $n \times n$ image with Parallel Stroked Multi Line is $O(n^3)$.*

PROOF. Based on lemma 2, the best approximation of each $m \times m$ patch is computed in $O(m^3)$. Without loss of generality, we assume that n is dyadic ($n = 2^J$). The proposed method is based on the adaptive geometrical wavelet. According to [47], when using quadtree for partitioning image, time complexity for the whole image, $\Psi(n)$, is computed as bellow:

$$\begin{aligned}
 \Psi(n) &= \sum_{j=0}^J 2^{2j} \times a \times (2^{J-j})^3 \\
 &= a \sum_{j=0}^J 2^{2j} 2^{3(J-j)} = a 2^{2J} \sum_{j=0}^J 2^j \\
 &= a n^2 (2^{J+1} - 1) = a(2n^3 - n^2) \\
 &= O(n^3),
 \end{aligned}$$

where a is a constant.

5.4. Model compression

Most compression algorithms like JPEG2000 use complex relationship among pixels to predict gray level value of a pixel. Thus, the redundancy between gray level values is reduced and the compression algorithms achieve lower bit rate. To have a fair comparison between the PSML algorithm and other compression methods, the redundancy in PSML coefficients must be reduced. In the proposed method, the gray level and structural parameters of each pixels are similar to the neighbor patches. So, a compression algorithm can be applied on these parameters. In this paper, a simple compression algorithm based on Low Complexity Lossless Compression for Images (LOCO-I) predictor and Golomb-Rice coding [56] is used to compress just the gray level coefficients obtained from the PSML model.

6. Experimental results

At first part of this section, the methods, tools and databases used for experiments are described. Then in second part, the experiments and parameters used for each one of them are explained. Also, a complete discussion on results obtained by the experiments is provided.

6.1. Settings and Methods

6.1.1. Compression methods

Nearly all the new methods proposed for compressing fingerprint images, compare their works with JPEG2000 method. Therefore, for sake of consistency, the PSML algorithm is compared against the JPEG2000. In this paper, implementation provided by Kakadu Software [57] is used. In addition we compare the PSML algorithm against Wedgelets Transform proposed by [47].

6.1.2. Biometric recognition systems

Among the well-known biometric recognition systems (e.g. VeriFinger, eFinger and NIST FIVB), we choose VeriFinger to use in this paper. VeriFinger is a commercial tool that acts based on minutiae matching. Experiments have shown that VeriFinger is more robust against noise and compression artifact compared to the other tools. VeriFinger feature matching algorithm provides similarity score as the result. The higher is the score, the higher is probability that feature collections are obtained from same person. The version used in this paper was VeriFinger SDK v7.1 published on 24/08/2015 [58].

6.1.3. Databases

to evaluate the proposed method, three fingerprint images databases are used that shortly named DB1, DB2 and DB3. These databases are described in the following:

DB1: This database is *DB1_B* from *Fingerprint Verification Competition (FVC2002)* that contains 10 fingers and 8 impressions per finger (80 fingerprints in all). Each fingerprint has 388×374 pixels obtained with 500 dpi resolution. The image with name '*101.1.tif*' is called PIC1.

DB2: This database is *DB4_B* from *Fingerprint Verification Competition (FVC2002)* that contains 10 fingers and 8 impressions per finger (80 fingerprints in all). Each fingerprint has 288×384 pixels obtained with about 500 dpi resolution. The image with name '*103.1.tif*' is called PIC2.

DB3: This database contains fingerprint images from 7 persons. Each person has data for all 10 fingers (except one that has data for 5 fingers) and 8 impressions exist per finger (520 fingerprints overall). These fingerprint obtained by *U.are.U 400* biometric hardware with 326×357 pixels, 500 ppi resolution and published by *NeuroTechnology* company. Due to the fact that fingerprint data of each finger is different from other fingers in same person, they could be assumed as 65 fingers and 8 impression per finger. The image with name '*012.1.2.tif*' is called PIC3.

6.2. Experiments

Experiment 1: All images in DB1, DB2 and DB3 are encoded with Wedgelets Transform and PSML algorithm at different compression rates. The settings used for both algorithms are 8 bits for gray level value, builds quadtree up to 7 levels of model and do not use compression on the resulting coefficients discussed on 5.4. According to the results, almost all images of all databases

produced the same results across different rates with PSNR measure. Fig. 5 shows compression result for PIC1, PIC2 and PIC3 obtained in compression rates from 0.01 to 4 bit per pixel (bpp). In this figure, two types of diagram are provided. The rate-distortion diagrams provide the better visualization of results in lower compression rates, while the compression ratio-distortion diagrams provide the better one in higher compression rates. In addition, Fig. 6 shows the compressed images of PIC1, PIC2 and PIC3 at two different rates, 1 bpp and 0.1 bpp, with both compression algorithms.

Corollary 1 (Wedgelets vs PSML): In experiment 1, both PSML algorithm and Wedgelets Transform used the same configurations. In lower compression rates, the percentage of nodes in the quadtree, which decorated with the model, is low and size of these nodes are small. So, the model has minimum influence in encoding rate. Also, because the PSML model takes more bits than the wedgelet model, images encoded with Wedgelets Transform has smaller size than the images encoded by the PSML algorithm in very low compression rates. This corollary is visible in right side of Fig. 5a. By increasing the compression rate, the percentage of nodes in the quadtree which decorated with the model is increased and size of these nodes become larger. Because the PSML model is more fitted to fingerprint structure than wedgelet model, in not so small node, the rate-distortion ratio made by this model has higher values. So, in this range of rates, the PSML algorithm has advantage Wedgelets Transform in PSNR value. Also, by increasing the compression rate, this advantage become more pronounced. This corollary is visible in Fig. 5. The noise and complexity in fingerprint images increases the percentage of nodes in quadtree decorated with the model. As the images of DB2 and DB3 have such a situation, in encoded images of these databases, the PSNR value results by PSML is higher than the PSNR value resulted by Wedgelets Transform. In Fig. 6, some example of encoded images at two different compression rates are shown. The encoded images by PSML algorithm has better visual quality and minutiae is better identifiable, specially at higher compression rates.

Experiment 2: All images in DB1, DB2 and DB3 are encoded with JPEG2000 and PSML algorithm at different compression rates ranging from 0.01 to 4 bpp. The PSML algorithm builds quadtree up to 7 levels for model. Also, it uses arbitrary bit rates from 3 to 8 bits for gray level values (i.e., the same bit rate for all value). Also, we apply the compression algorithm proposed in section 5.4. According to results, almost in all images, the same result was obtained. Similar to experiment 1, the rate-distortion and compression ratio-distortion diagrams for PIC1, PIC2 and PIC3 are shown in Fig. 7. In rate-distortion diagrams of this figure, PSML algorithm has some jumps, because the entire graylevel values in image is quantized with the same number of bits. So, when number of bits changed from 8 to 7, 7 to 6 or etc., it is observed as a jump in the PSNR value. Also, Fig. 8 shows the compressed images obtained by both algorithms at two different rates, 0.1 bpp and 1 bpp.

Experiment 3: In this experiment, the effect of compression algorithms examined with an AFIS tool. This experiment applied to DB3 images, examined PSML and JPEG2000 compression algorithms and repeated for various

number of compression rates. At each compression rate and for each compression algorithm, all images are encoded in specified rate. Then the VeriFinger tool is applied to all compressed images to obtain matching value between each two pair of images. Finally, for each compression rate and for each compression algorithm, the FAR/FRR diagram was obtained. From each diagram, the EER value can be extracted. Fig. 9 shows the mean EER value in different compression rates for both compression algorithms.

Corollary 2 (JPEG2000 vs PSML): The resulting diagrams of experiment 2 (Fig. 7), can be divided into three parts. In very low compression rates (compression ratio less than 3:1), the PSNR value of the PSML algorithm has advantage over JPEG2000 algorithm. This is because most nodes in quadtree are constant, so the coefficient compression algorithm acts like JPEG-LS. In middle range of compression rates (compression ratio greater than 3:1 and less than 80:1), the PSNR value of JPEG2000 algorithm has advantage over the PSML algorithm. But, as can be seen in Fig. 9, the mean EER value of JPEG2000 does not have this advantage. At compression rates less than 40:1, the mean EER value of both algorithms is approximately equal and at compression rates greater than 40:1, the PSML algorithm has advantage over JPEG2000 algorithm in terms of mean EER value. This implies that although the PSML algorithm does not preserve gray level value of each pixels, but it preserves the discriminating information required by the VeriFinger tool to compute the matching result between two fingerprint images. It is clear that the information is fingerprint structure and minutiae. In very high compression rate (compression ratio greater than 80:1), the PSNR value and mean EER value of PSML algorithm has great advantage over JPEG2000 algorithm. It implies that for higher compression rates, the PSML algorithm is a much better choice than JPEG2000 for compression fingerprint images.

Corollary 3 (PSML robustness): By analyzing the results of experiment 3 (Fig. 9), we conclude that the PSML algorithm is much more robust in terms of preserving fingerprint information than JPEG2000 algorithm. In common range of compression ratio (less than 40:1) neither PSML algorithm nor JPEG2000 has advantage over another one. But in higher compression rates, mean EER value of JPEG2000 is close to 50%, that means identification algorithm works like a random algorithm and the compressed images are not identifiable from each other. But, mean EER value of PSML algorithm remains in acceptable range and the compressed images identifiable from each other.

Corollary 4 (Fingerprint image compression in high compression ratio): At high compression ratios, fingerprint images encoded by JPEG2000 lose their structure and discriminating information. But, if they are compressed by PSML algorithm, the fingerprint discriminating information is better retained. Fig. 7 shows JPEG2000 is not have ability to encode fingerprint images in compression ratio greater than 200:1 and the PSNR value is fall sharply at these compression ratios. But, the quality of the compressed images by JPEG2000 decreases before this range of compression ratios. This corollary is confirmed by Fig. 10.

7. Conclusion and future works

A simple structure called PSML model is proposed. By using this model in compressing algorithm, the structure of fingerprint images are preserved even in very high compression rates. In comparison with the most known algorithm in its family, the experiments show that the PSML algorithm has better PSNR values in comparison with Wedgelets Transform.

One of the most important issues that must be considered when developing a compression algorithm for fingerprint images is preserving minutiae, which are used by AFIS systems for identification. In high compression rates, the JPEG2000 algorithm is ineffective, but the PSML algorithm can compress fingerprint images without losing structure and discriminating information of these images. According to the result of AFIS system applied on encoded images with PSML algorithm, the proposed method is preferred to JPEG2000 when high compression rates are needed.

There are several future directions that can follow this work. First, the compression algorithm that introduced in section 5.4 is very simple. It can be improved to achieve a better compression algorithm. As a result, the encoded images become more compressed while its advantages remain as before. Secondly, the proposed model could be extended to consider bifurcation and delta (Y-shaped ridge meeting) that are most important for minutiae extraction. Thirdly, other initialization methods can be investigated for better results. Fourthly, the fast approximation algorithm can be replaced either by an optimization algorithm that the solution is more closer to global optimal solution, or by a faster approximation algorithm. Finally, the proposed algorithm can be implemented in parallel to reduce the time complexity.

References

References

- [1] H. Kasban, Fingerprints verification based on their spectrum, *Neuro-computing* 171 (2016) 910 – 920. doi:<http://dx.doi.org/10.1016/j.neucom.2015.07.030>.
- [2] A. Morales, A. Kumar, M. A. Ferrer, Interdigital palm region for biometric identification, *Computer Vision and Image Understanding* 142 (2016) 125 – 133. doi:<http://dx.doi.org/10.1016/j.cviu.2015.09.004>.
- [3] H. Zhang, J. R. Beveridge, B. A. Draper, P. J. Phillips, On the effectiveness of soft biometrics for increasing face verification rates, *Computer Vision and Image Understanding* 137 (2015) 50 – 62. doi:<http://dx.doi.org/10.1016/j.cviu.2015.03.003>.
- [4] M. Eskandari, nsen Toygar, Selection of optimized features and weights on face-iris fusion using distance images, *Computer Vision and Image Understanding* 137 (2015) 63 – 75. doi:<http://dx.doi.org/10.1016/j.cviu.2015.02.011>.

- [5] M. R. Aqmar, Y. Fujihara, Y. Makihara, Y. Yagi, Gait recognition by fluctuations, *Computer Vision and Image Understanding* 126 (2014) 38 – 52. doi:<http://dx.doi.org/10.1016/j.cviu.2014.05.004>.
- [6] A. Drosou, D. Ioannidis, K. Moustakas, D. Tzovaras, Spatiotemporal analysis of human activities for biometric authentication, *Computer Vision and Image Understanding* 116 (3) (2012) 411 – 421, special issue on Semantic Understanding of Human Behaviors in Image Sequences. doi:<http://dx.doi.org/10.1016/j.cviu.2011.08.009>.
- [7] A. Mascher-kampfer, H. Stögner, A. Uhl, Comparison of compression algorithms impact on fingerprint and face recognition accuracy, in: *Visual Computing and Image Processing VCIP 07, Proceedings of SPIE*, Vol. 6508, 2007. doi:10.1117/12.699199.
- [8] D. Peralta, M. Galar, I. Triguero, D. Paternain, S. Garca, E. Barrenechea, J. M. Bentez, H. Bustince, F. Herrera, A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation, *Information Sciences* 315 (2015) 67 – 87. doi:<http://dx.doi.org/10.1016/j.ins.2015.04.013>.
- [9] J. Thärnå, K. Nilsson, J. Bigun, Orientation scanning to improve lossless compression of fingerprint images, in: J. Kittler, M. Nixon (Eds.), *Audio- and Video-Based Biometric Person Authentication*, Vol. 2688 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003, pp. 343–350. doi:10.1007/3-540-44887-X_41.
- [10] G. Shao, Y. Wu, Y. A, X. Liu, T. Guo, Fingerprint compression based on sparse representation, *IEEE Transactions on Image Processing* 23 (2) (2014) 489–501. doi:10.1109/TIP.2013.2287996.
- [11] W. Pennebaker, J. Mitchell, *JPEG Still Image Compression Standard*, Van Nostrand Reinhold, New York, NY, USA, 1993.
- [12] J. Bradley, C. Brislawn, The wavelet/scalar quantization compression standard for digital fingerprint images, in: *IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 3, 1994, pp. 205–208. doi:10.1109/ISCAS.1994.409142.
- [13] M. Marcellin, M. Gormish, A. Bilgin, M. Boliek, An overview of jpeg-2000, in: *Data Compression Conference (DCC) Proceedings*, 2000, pp. 523–541. doi:10.1109/DCC.2000.838192.
- [14] A. Skodras, C. Christopoulos, T. Ebrahimi, The jpeg 2000 still image compression standard, *Signal Processing Magazine, IEEE* 18 (5) (2001) 36–58. doi:10.1109/79.952804.
- [15] M. A. Lepley, Profile for 1000ppi fingerprint compression, Tech. Rep. MTR 04B0000022, The MITRE Corporation (April 2004).

- [16] R. C. Kidd, Comparison of wavelet scalar quantization and jpeg for fingerprint image compression, *Journal of Electronic Imaging* 4 (1) (1995) 31–39. doi:10.1117/12.195010.
- [17] A. Zhang, B. Cheng, R. Acharya, R. Menon, Comparison of wavelet transforms and fractal coding in texture-based image retrieval, in: *Proc. of SPIE: Visual Data Exploration and Analysis III*, SPIE, 1996, pp. 116–125.
- [18] D. Krämer, A. Bruckmann, T. Freina, M. Reichl, A. Uhl, Comparison of wavelet, fractal, and dct based methods on the compression of prediction-error images, in: *Proceedings of the International Picture Coding Symposium (PCS’97)*, volume 143 of *ITG-Fachberichte*, 1997, pp. 393–397.
- [19] T. Aziz, M. Haque, Comparative study of different fingerprint compression schemes, in: *Communications, Computers and signal Processing*, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on, 2005, pp. 562–565. doi:10.1109/PACRIM.2005.1517351.
- [20] J. M. Libert, S. Orandi, J. D. Grantham, Comparison of the wsq and jpeg 2000 image compression algorithms on 500 ppi fingerprint imagery : Nistir 7781, Tech. rep., National Institute of Standards and Technology, http://www.nist.gov/manuscript-publication-search.cfm?pub_id=910658 (April 2012).
- [21] J. M. Libert, J. D. Grantham, S. Orandi, A 1d spectral image validation/verification metric for fingerprints : Nistir 7599, Tech. rep., National Institute of Standards and Technology, http://www.nist.gov/manuscript-publication-search.cfm?pub_id=903078 (August 2009).
- [22] G. Weinhandel, H. Stogner, A. Uhl, Experimental study on lossless compression of biometric sample data, in: *Image and Signal Processing and Analysis*, 2009. ISPA 2009. Proceedings of 6th International Symposium on, 2009, pp. 517–522. doi:10.1109/ISPA.2009.5297682.
- [23] B. Sherlock, D. Monro, Optimized wavelets for fingerprint compression, in: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Proceedings*, Vol. 3, 1996, pp. 1447–1450 vol. 3. doi:10.1109/ICASSP.1996.543934.
- [24] B. Sherlock, D. Monro, Psychovisually tuned wavelet fingerprint compression, in: *Image Processing, 1996. Proceedings., International Conference on*, Vol. 1, 1996, pp. 585–588 vol.2. doi:10.1109/ICIP.1996.560929.
- [25] B. Babb, F. Moore, The best fingerprint compression standard yet, in: *Systems, Man and Cybernetics*, 2007. ISIC. IEEE International Conference on, 2007, pp. 2911–2916. doi:10.1109/ICSMC.2007.4414224.
- [26] K. Shanavaz, P. Mythili, Evolution of better wavelet coefficients for fingerprint image compression using cropped images, in: *Advances in Computing*

- and Communications (ICACC), 2012 International Conference on, 2012, pp. 126–129. doi:10.1109/ICACC.2012.28.
- [27] S. Kasaei, M. Deriche, B. Boashash, Fingerprint compression using a modified wavelet transform and pyramid lattice vector quantization, in: TENCON '96. Proceedings., 1996 IEEE TENCON. Digital Signal Processing Applications, Vol. 2, 1996, pp. 798–803 vol.2. doi:10.1109/TENCON.1996.608448.
 - [28] M. Deriche, S. Kasaei, A. Bouzerdoun, A novel fingerprint image compression technique using the wavelet transform and piecewise-uniform pyramid lattice vector quantisation, in: Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, Vol. 3, 1999, pp. 359–363 vol.3. doi:10.1109/ICIP.1999.817135.
 - [29] S. Kasaei, M. Deriche, B. Boashash, A novel fingerprint image compression technique using wavelets packets and pyramid lattice vector quantization, Image Processing, IEEE Transactions on 11 (12) (2002) 1365–1378. doi:10.1109/TIP.2002.802534.
 - [30] G. Khuwaja, A. Tolba, Fingerprint image compression, in: Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop, Vol. 2, 2000, pp. 517–526 vol.2. doi:10.1109/NNSP.2000.890129.
 - [31] Y.-L. Wang, C.-T. Liao, A. Su, S.-H. Lai, Fingerprint compression: An adaptive and fast dct-based approach, in: 17th IEEE International Conference on Image Processing (ICIP), 2010, pp. 3109–3112. doi:10.1109/ICIP.2010.5654009.
 - [32] A. Ouanane, A. Serir, Fingerprint compression by ridgelet transform, in: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2008, pp. 46–51. doi:10.1109/ISSPIT.2008.4775682.
 - [33] I. Ersoy, F. Ercal, M. Gokmen, A model-based approach for compression of fingerprint images, in: Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, Vol. 2, 1999, pp. 973–977 vol.2. doi:10.1109/ICIP.1999.823043.
 - [34] W. Chang, H. Soliman, A. Sung, Fingerprint image compression by a natural clustering neural network, in: Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, Vol. 2, 1994, pp. 341–345 vol.2. doi:10.1109/ICIP.1994.413588.
 - [35] A. Mohammed, R. Minhas, Q. Wu, M. Sid-Ahmed, Fingerprint image compression standard based on wave atoms decomposition and self organizing feature map, in: Electro/Information Technology, 2009. EIT '09. IEEE International Conference on, 2009, pp. 367–372. doi:10.1109/EIT.2009.5189644.

- [36] S. Zhao, X.-F. Wang, Fingerprint image compression based on directional filter banks and tcq, in: Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on, 2009, pp. 660–663. doi:10.1109/WKDD.2009.146.
- [37] A. Singla, R. Anand, A. Tiwari, V. Bajpai, Anovel prediction algorithm for lossless compression of fingerprint images, in: Signal Processing (ICSP), 2012 IEEE 11th International Conference on, Vol. 1, 2012, pp. 663–666. doi:10.1109/ICoSP.2012.6491575.
- [38] B. A. Olshausen, D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature* 381 (1996) 607–609. doi:10.1038/381607a0.
- [39] S. Mallat (Ed.), *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd Edition, Academic Press, Boston, 2009.
- [40] M. J. Mohlenkamp, M. C. Pereyra (Eds.), *Wavelets, Their Friends, and What They Can Do for You*, EMS Series of Lectures in Mathematics, European Mathematical Society (EMS), Zürich, 2008.
- [41] G. Welland (Ed.), *Beyond Wavelets*, Academic Press, San Diego, CA, 2003.
- [42] F. G. Meyer, R. R. Coifman, Brushlets: A tool for directional image analysis and image compression, *Applied and Computational Harmonic Analysis* 4 (2) (1997) 147 – 187. doi:http://dx.doi.org/10.1006/acha.1997.0208.
- [43] E. Candès, Ridgelets: Theory and applications, Ph.D. thesis, Department of Statistics, Stanford University, Stanford, USA (1998).
- [44] E. Candès, D. Donoho, Curvelets— a surprisingly effective nonadaptive representation for objects with edges, in: A. Cohen, C. Rabut, L. L. Schumaker (Eds.), *Curves and Surfaces Fitting*, Vanderbilt University Press, Saint-Malo, 1999.
- [45] M. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, *Image Processing, IEEE Transactions on* 14 (12) (2005) 2091–2106. doi:10.1109/TIP.2005.859376.
- [46] D. Labate, W.-Q. Lim, G. Kutyniok, G. Weiss, Sparse multidimensional representation using shearlets, in: M. Papadakis, A. F. Laine, M. A. Unser (Eds.), *Wavelets XI*, Vol. 5914 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2005, pp. 254–262. doi:10.1117/12.613494.
- [47] D. L. Donoho, Wedgelets: Nearly-minimax estimation of edges, *Annals of Statistics* 27 (1999) 859–897.

- [48] D. L. Donoho, X. Huo, Beamlet pyramids: a new form of multiresolution analysis suited for extracting lines, curves, and objects from very noisy image data, in: A. Aldroubi, A. F. Laine, M. A. Unser (Eds.), *Wavelet Applications in Signal and Image Processing VIII*, Vol. 4119 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2000, pp. 434–444. doi:10.1117/12.408630.
- [49] A. Lisowska, Second order wedgelets in image coding, in: *EUROCON*, 2007. The International Conference on "Computer as a Tool", 2007, pp. 237–244. doi:10.1109/EURCON.2007.4400239.
- [50] R. Willett, R. Nowak, Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging, *Medical Imaging, IEEE Transactions on* 22 (3) (2003) 332–350. doi:10.1109/TMI.2003.809622.
- [51] R. M. Willett, Multiresolution methods for recovering signals and sets from noisy observations, Ph.D. thesis, Rice University (April 2005).
- [52] V. Chandrasekaran, M. Wakin, D. Baron, R. Baraniuk, Surflets: a sparse representation for multidimensional functions containing smooth discontinuities, in: *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, Chicago, IL, 2004. doi:10.1109/ISIT.2004.1365602.
- [53] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, *Image Processing, IEEE Transactions on* 14 (4) (2005) 423–438. doi:10.1109/TIP.2005.843753.
- [54] S. Mallat, Geometrical grouplets, *Applied and Computational Harmonic Analysis* 26 (2) (2009) 161 – 180. doi:http://dx.doi.org/10.1016/j.acha.2008.03.004.
- [55] J. Krommweh, Tetrolet transform: A new adaptive haar wavelet algorithm for sparse image representation, *Journal of Visual Communication and Image Representation* 21 (4) (2010) 364 – 374. doi:http://dx.doi.org/10.1016/j.jvcir.2010.02.011.
- [56] R. F. Rice, Some practical universal noiseless coding techniques, Tech. Rep. JPL-PUBL-91-3, Nasa, Jet Propulsion Lab., California Inst. of Tech.; Pasadena, CA, United States (Nov. 1991).
URL <http://ntrs.nasa.gov/search.jsp?R=19920005393>
- [57] A. David Taubman, the University of New South Wales, Kakadu v7.7: A foundation for jpeg2000 applications (2015) [cited 12 November 2015].
URL <http://kakadusoftware.com>
- [58] Neurotechnology, Verifinger sdk, fingerprint identification for stand-alone or Web solutions (2015) [cited 12 November 2015].
URL <http://www.neurotechnology.com/verifinger.html>

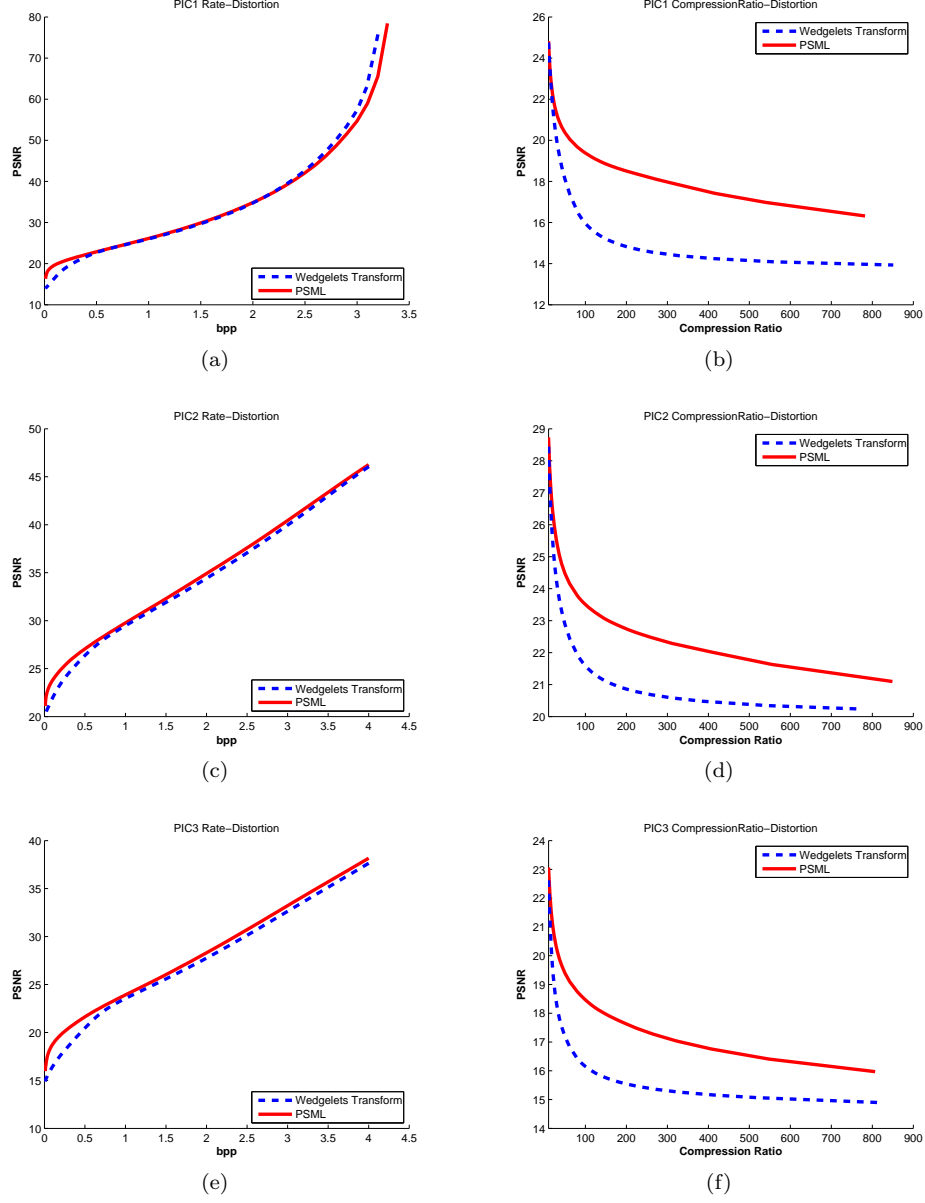


Figure 5: Compare rate-distortion (left) and compression ratio-distortion (right) for Wedgelets Transform and PSML algorithm according to experiment 1. Rate-distortion diagram shows results of low compression rates in detail, while compression ratio-distortion diagram shows result of high compression rates in detail. (a,b) PIC1 (c,d) PIC2 (e,f) PIC3.

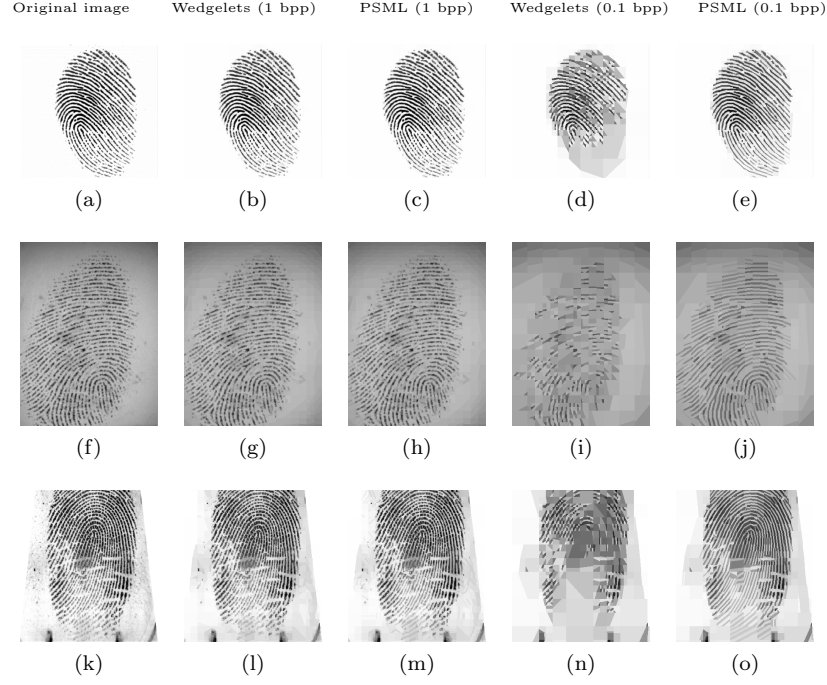


Figure 6: Samples of fingerprint images encoded by Wedgelets Transform and PSML algorithm, according to Experiment 1. First column shows the original fingerprint image. Second and fourth columns show the compressed fingerprint images by Wedgelets Transform. Third and Fifth columns show the compressed fingerprint images by PSML algorithm. Second and Third columns are encoded in 1 bpp, while fourth and fifth column are encoded in 0.1 bpp. First row belongs to result of experiment 1 on PIC1, while second and third rows belongs to PIC2 and PIC3, respectively. The PSNR value of images are (b,c) 26.03 , 26.15 (d,e) 16.51 , 19.69 (g,h) 29.49 , 29.78 (i,j) 21.94 , 23.78 (l,m) 23.54 , 23.92 (n,o) 16.43 ,18.76.

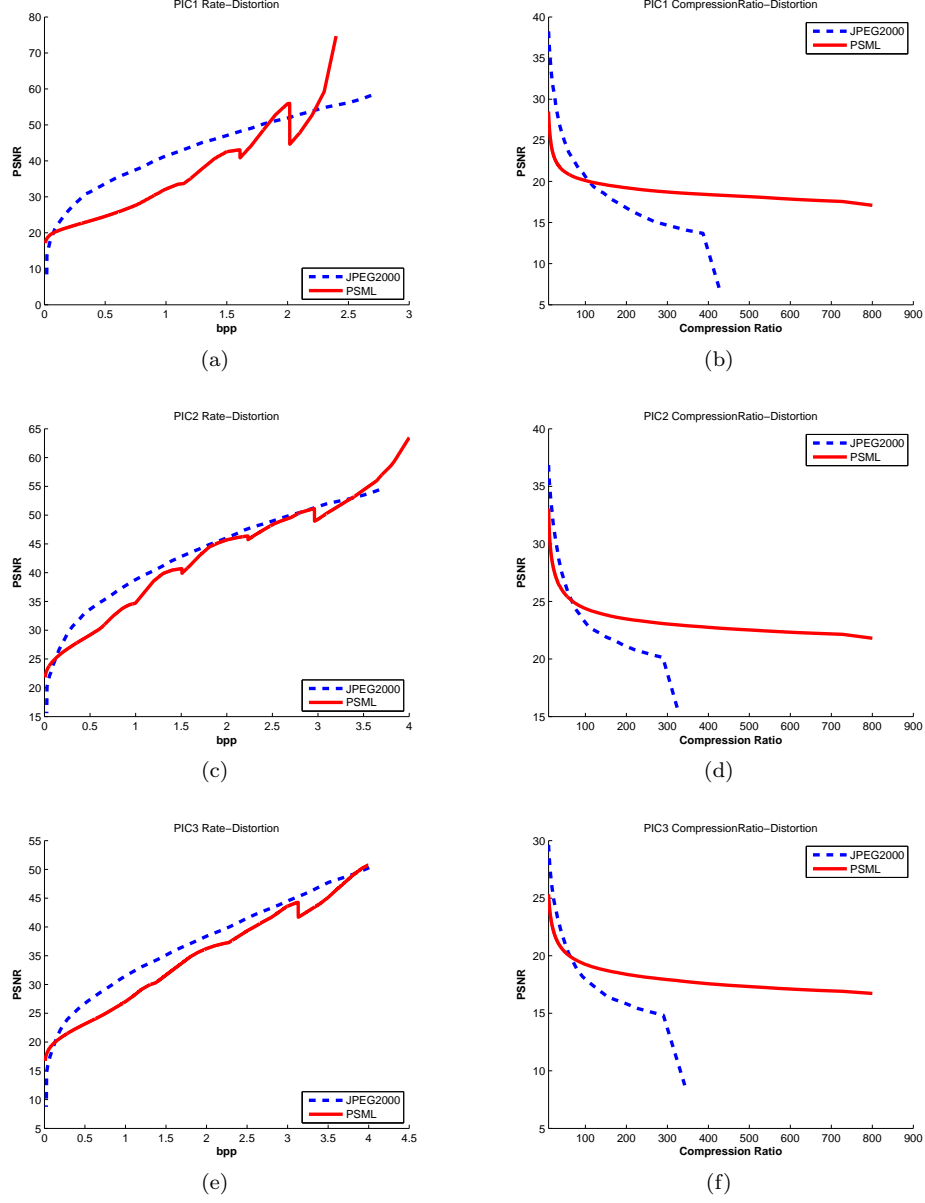


Figure 7: Compare rate-distortion (left) and compression ratio-distortion (right) for JPEG2000 algorithm and PSML algorithm on PIC3 according to experiment 2. Rate-distortion diagram shows results of low compression rates in detail, while compression ratio-distortion diagram shows result of high compression rates in detail. (a,b) PIC1 (c,d) PIC2 (e,f) PIC3.

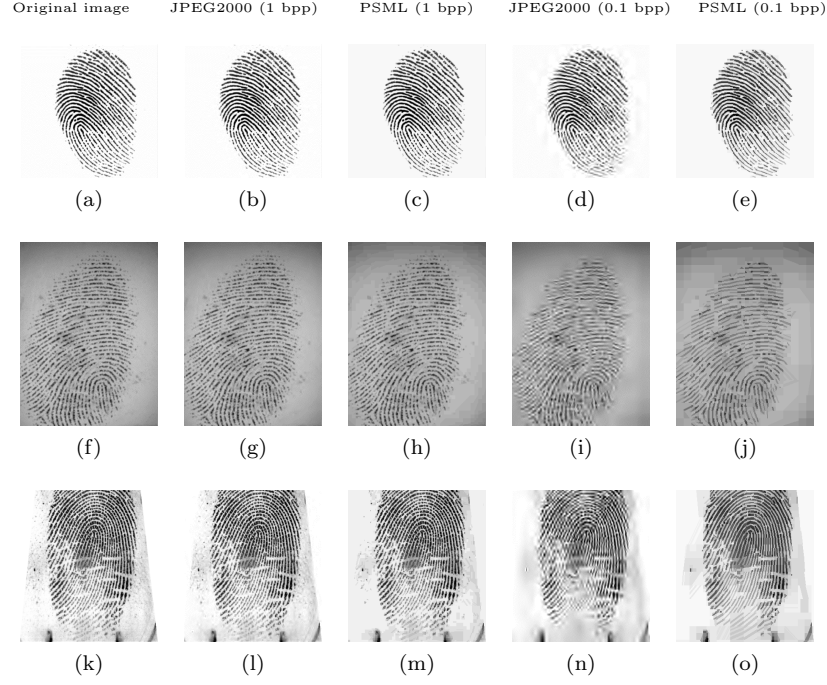


Figure 8: Samples of fingerprint images encoded by JPEG2000 and PSML algorithm, according to Experiment 2. First column shows the original fingerprint image. Second and fourth columns show the compressed fingerprint images by JPEG2000 algorithm. Third and Fifth columns show the compressed fingerprint images by PSML algorithm. Second and Third columns are encoded in 1 bpp, while fourth and fifth column are encoded in 0.1 bpp. First row belongs to result of experiment 1 on PIC1, while second and third rows belongs to PIC2 and PIC3, respectively. The PSNR value of images are (b,c) 41.36 , 32.19 (d,e) 21.90 , 20.41 (g,h) 38.79 , 34.74 (i,j) 24.22 , 24.75 (l,m) 31.49 , 27.03 (n,o) 18.73 , 19.58.

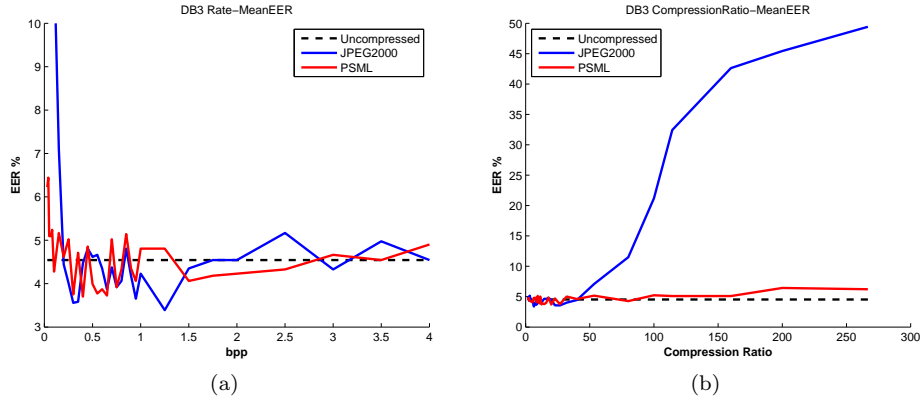


Figure 9: The EER values obtained by experiment 3 for various compression rates. The dashed line show the EER value of uncompressed images. The thinner and thicker lines show EER values of JPEG2000 and PSML algorithms, respectively.



Figure 10: Samples of compressed fingerprint images in high compression rates. (a) PIC1 original image (b,d) encoded with JPEG2000 (c,e) encoded with PSML. Compression ratio is (b,c) 178:1 (d,e) 320:1.